Claims

[c1] A method of sharing a state between stateful firewalls on a Multiple Entry/Exit Point (MEP) network for data exchange between a server and a client through firewalls physically remote from each other, comprising the steps of:
(a) one of the firewalls receiving an SYN packet sent from the client to the server;
(b) the firewall creating a modified SYN cookie (hereinafter referred to as an m.SYN cookie), modifying the SYN packet using the m.SYN cookie and sending the SYN packet to the server, and the server sending a SYN/ACK packet to the client in response to the SYN packet;
(c) the firewall, which has received the SYN/ACK packet, extracting a firewall identifier ID_{fw} from the SYN/ACK packet and sending the SYN/ACK packet to a corresponding one of the firewalls, the corresponding firewall searching a state table for connection information and sending the connection information, together with the SYN/ACK packet, to the firewall, which has received the SYN/ACK packet; and
(d) the firewall, which has re-received the SYN/ACK packet, updating the state table, changing an acknowl-

edgement number of the SYN/ACK packet to an Initial Sequence Number (ISN_c) + 1, and sending the SYN/ACK packet to the client.

[c2] The method as set forth in claim 1, wherein the firewalls share a synchronized time counter, which is increased at regular intervals, and a same secret key.

[c3] The method as set forth in claim 1, wherein the state table includes a difference between the ISN and the m.SYN cookie, and connection information, including a source address, a destination address, a protocol, a source port and a destination port number of the packet.

[c4] The method as set forth in claim 1, where step (a) further comprises the step of:
the firewall, which has received the SYN packet, inspecting the SYN packet according to a preset firewall rule, and performing step (b) if a current connection is a permitted connection, or discarding the SYN packet if the current connection is not the permitted connection.

[c5] The method as set forth in claim 2, wherein the m.SYN cookie includes upper bits of the ISN of the SYN packet, bits of time indicated by the time counter of the firewall, which creates the m.SYN cookie, at a time of creation of the m.SYN cookie, and bits of an output value of a hash

function.

[c6] The method as set forth in claim 2, wherein the m.SYN cookie includes ISN_{17}, T_0 and $Hash_{13} + ID_{fw}$, ISN_{17} being determined by upper 17 bits of the ISN of the SYN packet, T_0 being determined by least significant two bits of time indicated by the time counter of the firewall, which creates the m.SYN cookie, at the time of creation of the m.SYN cookie, $Hash_{13}$ being determined by the following Equation:

$Hash_{13}$ = Hash(k, sa, sp, da, dp, $time_{org}$, ISN_c>>15)%2^13

where Hash() is an output value of a hash function, k is a secret key, sa is a source address, sp is a source port number, da is a destination address, dp is a destination port number, ISN_c>>15 is a value obtained by eliminating lower 15 bits from ISN_c, Hash() % 2^13 is a value of lower 13 bits of the output value of the hash function, $time_{org}$ is time indicated by the time counter of the firewall, which creates the m.SYN cookie, at the time of creation of the m.SYN cookie

[c7] The method as set forth in claim 1, wherein step (b) is performed in such a way that the ISN of the SYN packet is replaced with the created m.SYN cookie, and the connection information including the difference between the ISN and the m.SYN cookie is stored in the state table of

the firewall.

[c8] The method as set forth in claim 1, wherein step (c) further comprises the steps of:
(c1) extracting the ID_{fw} from the SYN/ACK packet;
(c2) verifying whether the extracted ID_{fw} is valid;
(c3) comparing the ID_{fw}, which is verified to be valid at step (c2), with an ID_{fw} of the firewall, which has received the SYN/ACK packet; and
(c4) if, as a result of the comparison at step (c3), the two ID_{fw}s are identical with each other, searching the state table of the firewall that has received the SYN/ACK packet and modifying the state table and the SYN/ACK packet, or if the ID_{fw}s are different from each other, sending the SYN/ACK packet to the firewall corresponding to the extracted ID_{fw}.

[c9] The method as set forth in claim 8, wherein step (c1) is performed in such a way that the m.SYN cookie included in the SYN/ACK packet is extracted, and the ID_{fw} is extracted from the m.SYN cookie using the following equations.
ID_{fw} = (SC - Hash(k, sa, sp, da, dp, $time_{input}$, SC>>15))% 2^13
where SC is the m.SYN cookie included in the SYN/ACK packet, Hash() is an output value of a hash function, k is a secret key, sa is a source address, sp is a source port

number, da is a destination address, dp is a destination port number, $time_{input}$ is time obtained using the following Equation, SC>>15 is a value obtained by eliminating lower 15 bits from the SC, and () % 2^13 is a value of lower 13 bits of the value of ()

$time_{input} = time_{curr} + 1 ((time_{curr} + 1 - T_0) mod4)$

where $time_{curr}$ is the time indicated by the time counter of the firewall, which verifies the extracted m.SYN cookie, at the time of verification of the extracted m.SYN cookie, and T_0 is the least significant two bits of time indicated by the time counter of the firewall, which creates the m.SYN cookie, at the time of creation of the m.SYN cookie.

[c10] The method as set forth in claim 8, wherein step (c2) is performed in such a way as to compare the extracted ID_{fw} with a preset maximum ID_{fw}, and if the extracted ID_{fw} is not larger than the preset maximum ID_{fw}, verifying the extracted ID_{fw} to be valid, or if the extracted ID_{fw} is larger than the preset maximum ID_{fw}, verifying the extracted ID_{fw} to be invalid.